What is claimed is:

1. A method for double buffering serial transfers during a write operation, comprising:

(a) serially transferring address bits of a word into an address shift register,

(b) serially transferring data bits of the word into a data shift register;

(c) after completing the serial transfer of the address bits into the address shift register, transferring, in parallel, the address bits into an address holding register; and

(d) after completing the serial transfer of the data bits into the data shift register, transferring, in parallel, the data bits into a data holding register;

wherein after completing the parallel transfer of the address bits and data bits from the address and data shift registers to the address and data holding registers, the address and data shift registers are available to serially receive additional address bits and data bits of an additional word.

2. The method of claim 1, wherein:

step (a) include serially transferring one of the address bits per clock cycle;

step (b) include serially transferring one of the data bits per clock cycle;

step (c) includes transferring, in parallel, the address bits into the address holding register in one clock cycle or less; and

step (d) includes transferring, in parallel, the data bits into the data holding register in one clock cycle or less.

3. The method of claim 1, wherein the parallel transfer the address bits in step (c) is performed before the serial transfer of the data bits in step (b) is complete.

4. The method of claim 1, wherein step (c) is performed before step (d).

5. The method of claim 1, wherein steps (c) and (d) are performed simultaneously.

6. The method of claim 1, wherein the address and data holding registers each comprise a latch.

7. The method of claim 1, wherein the word is received over a serial line from a host.

8. The method of claim 1, wherein the transferring of the address bits and the data bits to the address and data holding registers in steps (c) and (d) is over parallel lines.

9. The method of claim 1, wherein the address bits specify a location to which to write the data bits.

10. A method for double buffering serial transfers during a read operation in which a host attempts to read data from a location in a device, comprising:

(a) serially transferring address bits received from the host into an address shift register in the device,

(b) serially transferring data bits present in a data shift register in the device, from the device to the host, wherein the data bits present in the data shift register are associated with a previous read operation;

(c) after completing the serial transfer of the address bits into the address shift register, transferring, in parallel, the address bits into an address holding register in the device, wherein the address bits identify the location, which contains requested data bits; and

(d) after the requested data bits are read from the location into a data holding register, transferring the requested data bits, in parallel, from the data holding register to the data shift register;

wherein the requested data bits will be transferred, serially, from the data shift register to the host the next time the host performs a read operation.

11. A double buffering system for use in a device to which a host writes data, and from which the host reads data, comprising:

an address shift register to serially receive address bits from a host, the address bits identifying a location to which to write data bits, or from which to read data bits;

a data shift register, to serially receive data from the host during a write operation, and to serially transfer data to the host during a read operation;

an address holding register to receive a parallel transfer of address bits from the address shift register; and

a data holding register, to receive a parallel transfer of data bits from the data shift register during a write operation, and to transfer in parallel data bits to the data shift register during a read operation.

12. The system of claim 11, wherein during a write operation, after the parallel transfers of the address bits and data bits from the address and data shift registers to the address and data holding registers, the address and data shift registers are available to serially receive additional address bits and data bits from the host.

13. The system of claim 12, wherein during a read operation, data bits transferred from the data holding register to the host, are associated with a previous read operation.

14. The system of claim 11, further comprising a serial controller to determine, based on a mode bit, whether a serial transfer from the host to the device is associated with a read operation or a write operation.

15. The system of claim 14, wherein the serial controller includes a counter that receives a clock signal, and wherein the serial controller uses the counter to determine which bits are address bits and which bits are data bits.

16. The system of claim 14, wherein the serial controller includes a switch to select between transferring bits to the first shift register and the second shift register.

17. A laser driver including:

a serial controller to control serial transfers between the laser driver and a host;

an address shift register to serially receive address bits from a host, the address bits identifying a location to which to write data bits, or from which to read data bits;

a data shift register, to serially receive data bit from the host during a write operation, and to serially transfer data bits to the host during a read operation,

an address holding register to receive a parallel transfer of address bits from the address shift register; and

a data holding register, to receive a parallel transfer of data bits from the data shift register during a write operation, and to transfer in parallel data bits to the data shift register during a read operation.

18. The system of claim 17, wherein during a write operation, after the parallel transfers of the address bits and data bits from the address and data shift registers to the address and data holding registers, the address and data shift registers are available to serially receive additional address bits and data bits from the host.

19. The system of claim 18, wherein during a read operation, data bits transferred from the data holding register to the host, are associated with a previous read operation.

20. The laser driver of claim 17, wherein the serial controller includes a counter that receives a serial clock (SCLK) signal, and wherein the serial controller uses the counter to determine which bits are address bits and which bits are data bits.

21. The laser driver of claim 20, wherein, during a write operation, the serial controller includes a switch to select between transferring bits to the address shift register and the data shift register.

22. The laser driver of claim 17, further comprising:

a parallel address bus connecting the address holding register to a register bank; and

a parallel data bus connecting the data holding register to the register bank.

23. The laser driver of claim 17, further comprising:

a parallel address bus connecting the address holding register to a timing memory; and

a parallel data bus connecting the data holding register to the timing memory.

24. The laser driver of claim 17, wherein the address and data holding registers each comprise a latch.

25. The laser driver of claim 20, further comprising logic that receives at least two least significant bits from the counter to thereby provide a level of confidence that a serial enable signal going high was not due to a glitch.

26. The laser driver of claim 25, wherein the logic includes:

an AND gate that receives the at least two least significant bits from the counter;

a 1-bit latch that includes a data input that receives an output from the AND gate, and an enable input attached to a send enable (SEN) line; and

a strobe circuit including an enable input that receives an output of the 1-bit latch;

wherein an output of the strobe circuit is used to ensure that possible glitches on the SEN line do not cause accidental reads or writes.